Statistic Multiplexed Computing – A Paradigm Shift Towards Data Parallel Tuple Space Computing

Justin Y. Shi | shi@temple.edu

Temple University

CnC-2014: The Sixth Annual **Concurrent Collections** Workshop **| September 18-19, 2014 | Hillsboro, Oregon**

What is Statistic Multiplexing?

- The least understood application programming discipline: **timeout handling**.
  - A novice application programmer would just issue an error message.
  - A experienced application programmer would try custom retransmission protocol.
- **Fact**: regardless the complexity of custom retransmission protocol, there can be no statistically significant improvement unless every retransmission takes an independent path.
- The dynamic path switching capability is called **Statistic Multiplexing. (think about TCP/IP)**

The Reproducibility Challenges

- The 2009 Yale University Roundtable meeting raised the reproducibility challenge of scientific research due to a number of high-profile recalled projects.
- Scientific research results must be cross-verified before moving forward to larger scale investigation/implementation.
- Large scale scientific computation results are difficult to reproduce but are needed for cross verification.
- Only 17 out of 67 U.S. nationally funded projects could be independently verified. Why?

Dependencies are The Enemies

- Experiences in large scale scientific computing projects showed that software and hardware dependencies have made independent reproduction of scientific research efforts very difficult.
- We found that tightly coupled, static program-processor binding is responsible for application scalability limit and the impossibility of reproduction of extreme scale applications.
- However, eliminating dependency is only the necessary condition, not sufficient condition for reproducibility.
- Q: In CnC, if dataflow is assumed, why still allow programmer to specify execution orders?

Problems Brought by the Weakest Link

The "virtual circuit" vulnerability:

- In the example OSI 7-layer model, layers 1-4 can scale infinitely for reliable packet transmissions.
- Layers 5-7 are vulnerable to any transient software and hardware failures.
- When combined with the lack of retransmission discipline at the application layer, an application will fail on ANY transient software or hardware failure.

Vulnerability

Protected for extreme scales

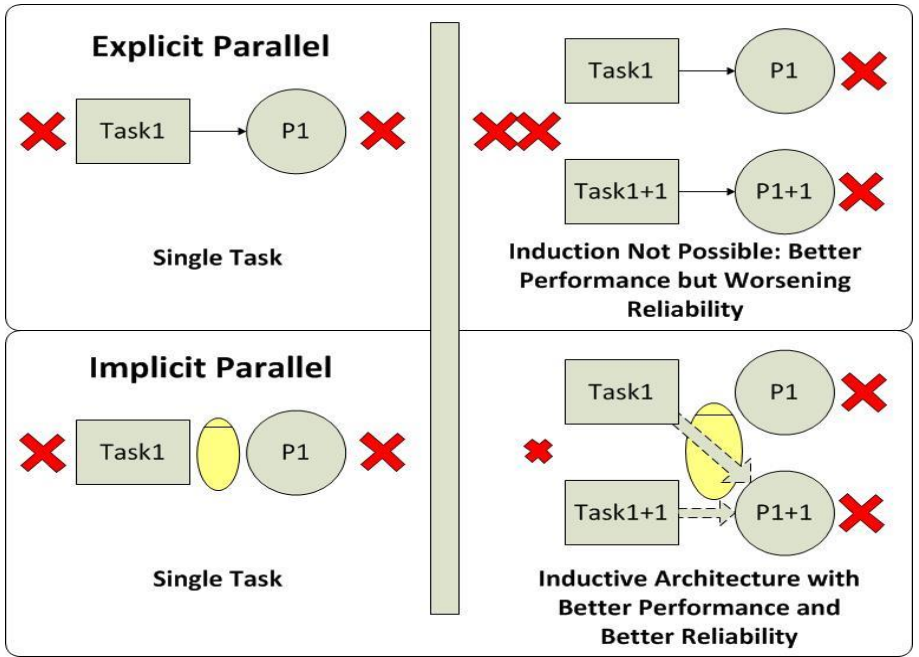
OSI Model			
	Data unit	Layer	Function
Host layers	Data	7. Application	Network process to application
		6. Presentation	Data representation, encryption and decryption, convert machine dependent data to machine independent data
		5. Session	Interhost communication, managing sessions between applications
	Segments	4. Transport	End-to-end connections, reliability and flow control
Media layers	Packet/Datagram	3. Network	Path determination and logical addressing
	Frame	2. Data link	Physical addressing
	Bit	1. Physical	Media, signal and binary transmission

Reproducibility and Scalability

- If we define a scalable application is an application it can gain performance and reliability at the same time by upscaling its processing infrastructure, then Reproducibility is a “necessary” condition for scalability, and
- Scalability is both necessary and sufficient conditions for reproducibility, because a scalable application is always more reproducible than a non-scalable application.

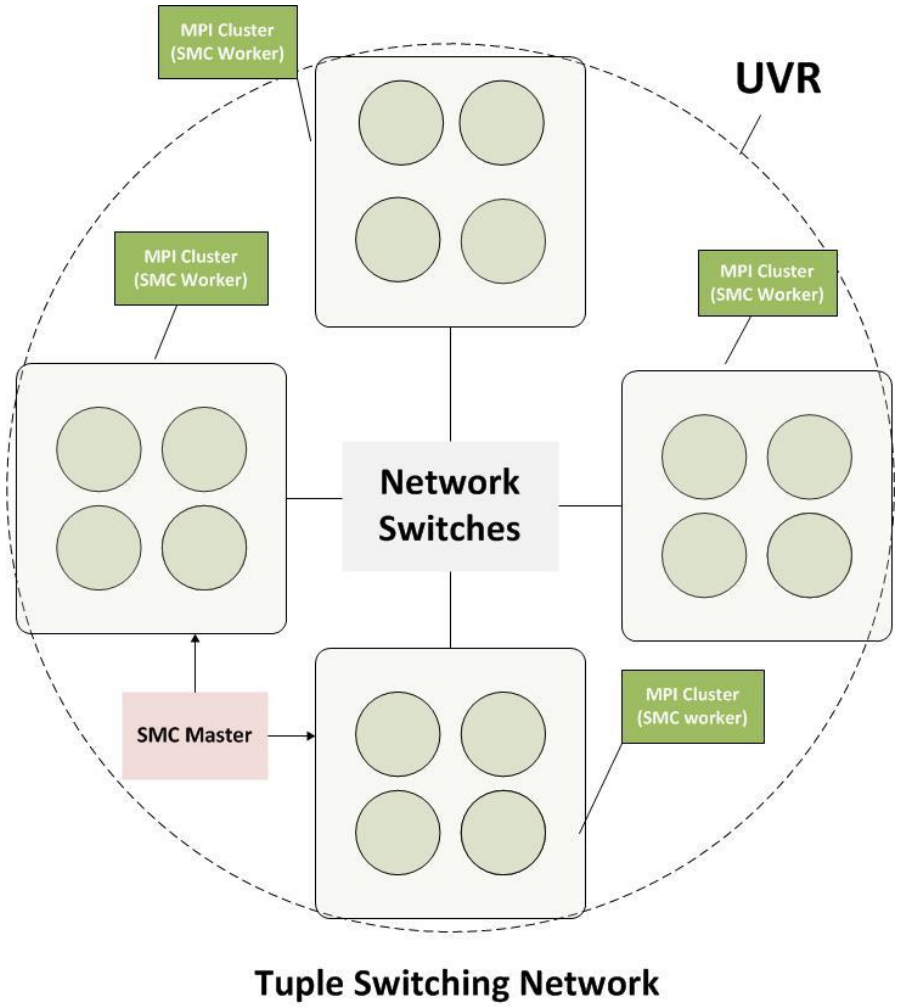
Why **Decoupling** can Solve Application Scalability Problem?

- **Obvious**: Parallel processors and their API evolve faster than the scientific research communities can manage. Decoupling allows more stable API platform -> better reproducibility of scientific research.
- **Not obvious**: Decoupled (implicit) parallel programs allow dynamic resource optimizations. Better reproducibility → unlimited scalability.



Inductive Architecture Possible

Tuple Switching Network



[Shi, et al, 2012]

<key, value> Data Parallel Processing (SMC)

- Statistic multiplexable at runtime (**tuple switching network**)
- Fills in the **missing discipline** in application data retransmission
- Inductive application architecture == **unlimited scalability**
- **Optimizable** for different processing environments (tuning granularity) (see next slide)
- **Easier** to program than explicit-parallel (minimal software and hardware dependencies)
- **Acceptable** to all applications (like Hadoop and Spark)

Why Statistic Multiplexed Computing (SMC)?

- A myth in application programming is the lack of timeout retransmission discipline. Without it, it is impossible to deliver lossless robust data communications [Alan Fekete, Nancy Lynch and others 1993].
- Tuple Switching Network provides statistic multiplexed resources for every tuple transmission and processing.
- Unlike traditional supercomputers, increasing the counts of processors and interconnects allows incrementally better application performance and reliability at the same time.

The New Performance Dimensions

- The implicit parallel paradigms dictate a more costly communication infrastructure compared to explicit parallel paradigms: each program-program communication overhead is almost doubled.
- But the decoupling effect liberates the dynamic granularity tuning at runtime – a long forbidden dimension of parallel performance optimization.
- Processing granularity tuning enables a parallel **completion time equilibrium** when the synchronization overhead is near zero.
- This enables implicit parallel programs to outperform explicit parallel programs, and quantitative scalability analysis/predictions.

Quantitative Scalability Analysis

- Given N, P, processing complexities Cs(N), Cp(N,P), processing and communication powers (w and u), we can reliably predict potential speedups under different conditions.
- Traditional Amdahl's Law and Gustafson's Law are indefinite laws that cannot reliably predict parallel application scalabilities [Shi, 1996].

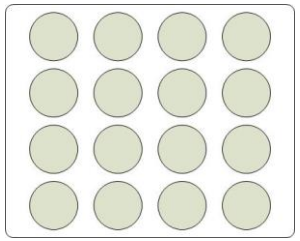
Unlimited Scalability and Diminishing Returns

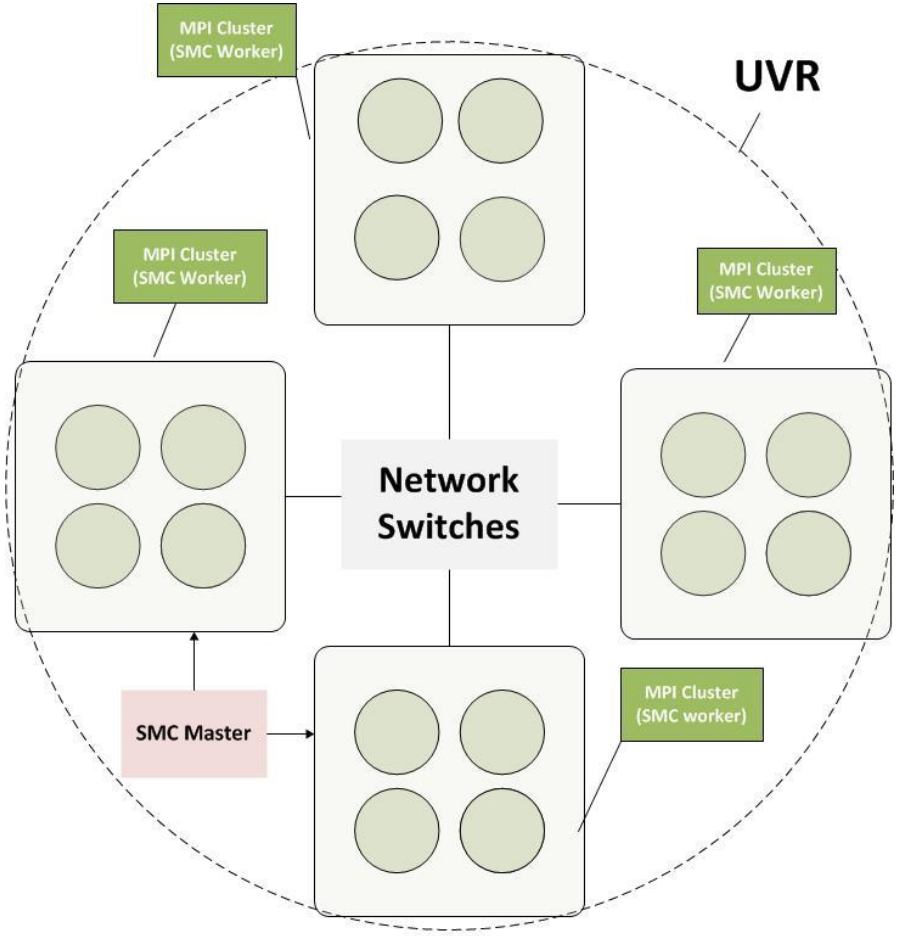
- Diminishing of return is the law of economics.
- Unlimited scalability is the feature of SMC applications.
- Both must co-exist.

A Practical Challenge

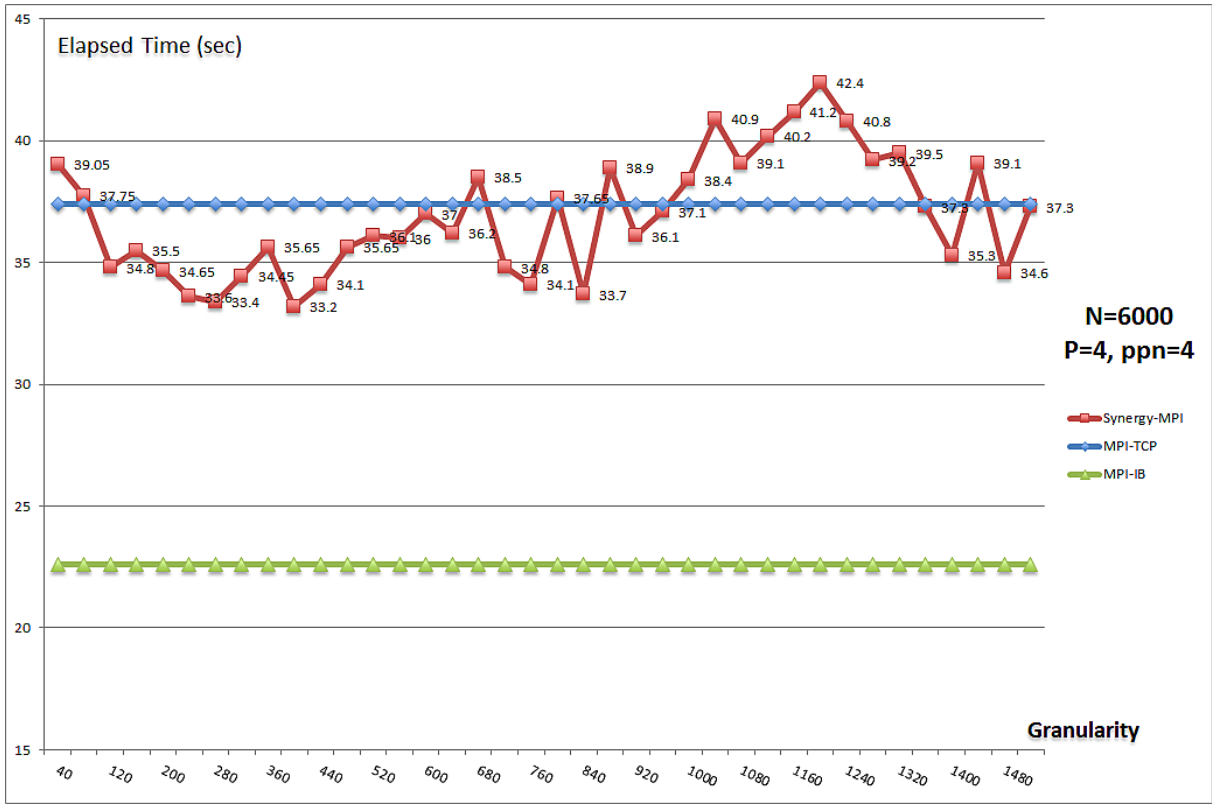
- Almost all existing parallel applications are explicit parallel programs.
- Rewriting these applications is a cost prohibitive proposition.
- The Internet provides an inspiration: SMC wrapping.
- The idea is to statistic multiplex multiple optimized smaller scale computation clusters in real time for incrementally better performance and reliability at the same time – A “Dod Fooding” idea.

Preliminary Results (SMC Wrapped MPI)



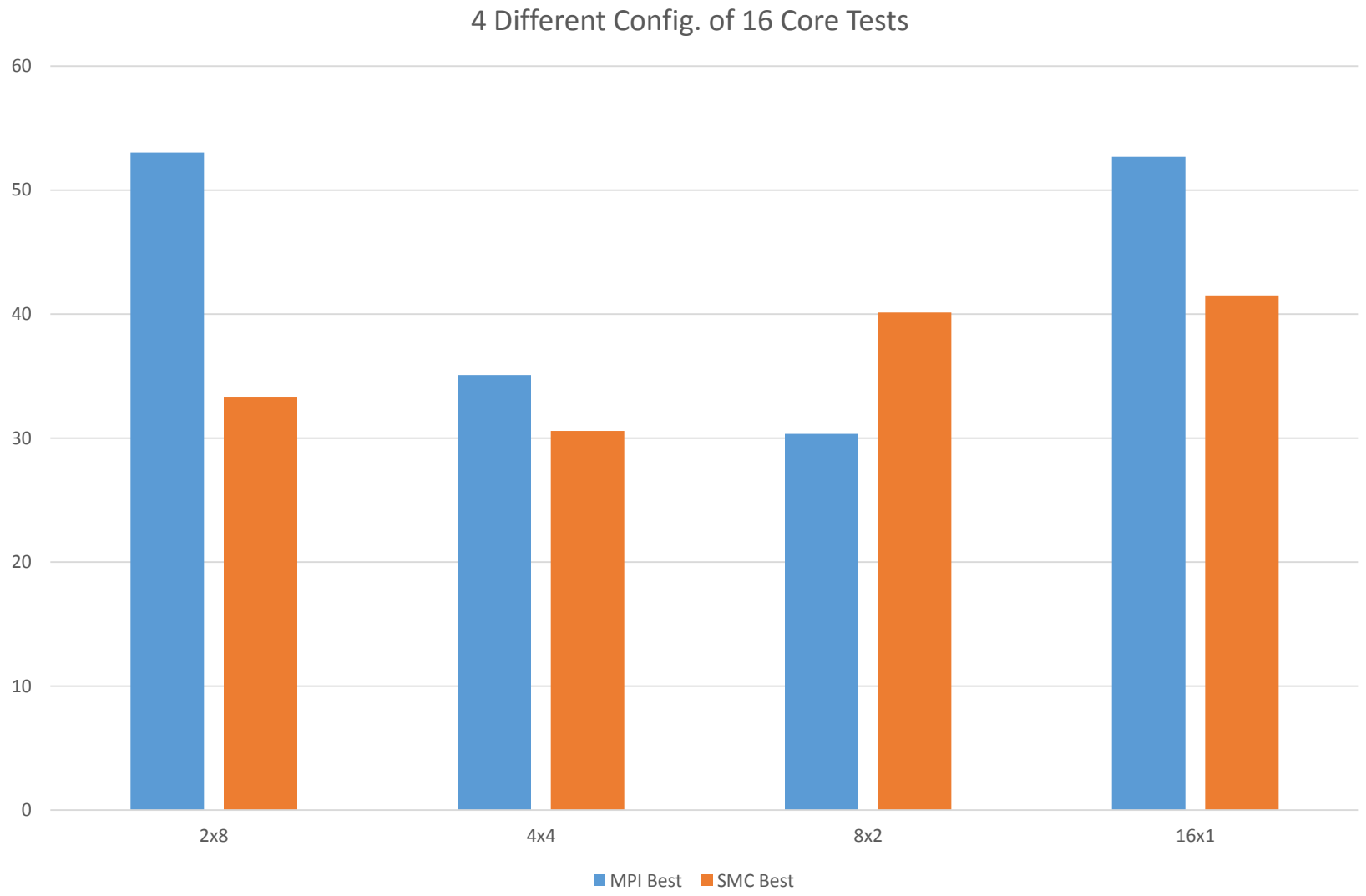


Tuple Switching Network



Questions and Software Requests: shi@temple.edu

More Recent Results



Conclusions

- Computational scientific research reproducibility is a necessary condition for scalability.
- A scalable application is always better reproducible than non-scalable application.
- A scalable application must be implicit parallel and SMC or otherwise it cannot afford the inductive architecture.
- A scalable application does not have structural inflection points.
- A SMC application can be optimized to outperform explicit parallel programs.
- A scalable application can be reliably quantified for potential speedups.
- Big data processing is better off with SMC parallel processing.
- Legacy applications can be re-engineered (wrapped) to meet the 21st century big data processing needs.

Acknowledgements

- National Science Foundation (NSF) grant CNS0958854
- Multiple NSF travel grants
- Industry sponsors: IBM T. J. Watson Research Center, Digital Equipment Corporation, Investors of Parallel Computers Technology Inc.
- Mr. Chris Gali of Adminovate Inc.